

UNITED STATES PATENT APPLICATION

for

A METHOD FOR GENERATING A SIMULATED NETWORK BASED ON AN
ACTUAL MANAGED NETWORK

Inventors:

Jeffrey B. Feldstein
Manuel Diaz
Tarun Raison

prepared by:

WAGNER, MURABITO & HAO L.L.P.
Two North Market Street
Third Floor
San Jose, CA 95113
(408) 938-9060

A METHOD FOR GENERATING A SIMULATED NETWORK BASED ON AN ACTUAL MANAGED NETWORK

FIELD OF INVENTION

5 The present invention relates to the field of computer network simulation. Specifically, the present invention relates to a method for generating a simulated network based on an actual managed network.

BACKGROUND OF THE INVENTION

10 In a large enterprise with a large number of computing devices, the networking infrastructure can be extraordinarily complex. For example, a network of a large enterprise may include five to ten thousand network devices as well as thousands of end-stations or "hosts". Network management systems are typically employed to monitor the network in order to ensure that
15 the network is functioning properly. CiscoWorks by Cisco Systems is an example of a network management system.

 In order to ensure proper functioning of a network management system, it is necessary to test the network management system while connected to a
20 network. In the past, the network management system would often be connected to a network of a live customer enterprise to conduct these tests. However, due to security and network disruption concerns, enterprises are no longer open to this sort of testing.

Currently, one way to test a network management system is to set up an actual network in a laboratory. For example, a network of ten to one hundred devices can be set up, and the network management system is used for monitoring the network. This method assumes that if the network management system works for a network of up to one hundred devices, it will work for a network of several thousand devices. However, this assumption is not always valid, as network complexity gets much greater as the number of devices increases.

10

Another way for testing a network management system is to connect the network management system to a simulated network. A simulated network is typically a data file and software system that is able to convince the network management system that it is connected to and monitoring an actual network.

15 Network simulation tools exist for assisting a user in creating a simulated network. Specifically, typical network simulation tools allow a user to specify devices of the simulated network. However, it is necessary for the user to manually enter in connection information for each device. That is, a configuration file that identifies the connections between the devices must be
20 created or edited for each device. For simulating a large network of several thousand devices this is very time consuming, thereby consuming substantial human and financial resources of the enterprise.

SUMMARY OF THE INVENTION

Various embodiments of the present invention, a method for generating a simulated network, are described herein. In one embodiment, a network discovery database for a managed network is accessed, wherein the network
5 discovery database comprises device information and connectivity information for a plurality of devices of the managed network. A build file describing the simulated network is automatically generated based on the network discovery database. In one embodiment, the build file is for use by a network management simulator to simulate a network.

10

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention:

5

FIGURE 1 is a block diagram of an exemplary computer system platform upon which embodiments of the present invention may be practiced.

FIGURE 2 is a block diagram of one embodiment of a system upon
10 which the present invention may be practiced.

FIGURES 3A and 3B are a flow chart illustrating a method for generating a simulated network, in accordance with one embodiment of the present invention.

15

FIGURES 4A and 4B are tables of an exemplary build file, in accordance with one embodiment of the present invention.

20

DETAILED DESCRIPTION

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings.

While the invention will be described in conjunction with the preferred

5 embodiments, it will be understood that they are not intended to limit the invention to these embodiments. On the contrary, the invention is intended to cover alternatives, modifications and equivalents, which may be included within the spirit and the scope of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention,

10 numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, components, structures and devices have not been described in detail so as to

15 avoid unnecessarily obscuring aspects of the present invention.

Various embodiments of the present invention, a method for generating a simulated network based on a managed network, are presented herein.

Embodiments of the present invention provide for accessing a network

20 discovery database for the managed network, and generating a simulated network based on the network discovery database. The simulated network is automatically generated based on the devices and the connections as described in the network discovery database.

Some portions of the detailed descriptions which follow are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here and generally conceived to be a self-consistent sequence of steps of instructions leading to a desired result. The steps are those requiring physical manipulations of data representing physical quantities to achieve tangible and useful results. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussions, it is appreciated that throughout the present invention, discussions utilizing terms such as “accessing”, “generating”, “extracting”, “determining”, “selecting”, “creating”, or the like, refer to the actions and processes of a computer system or similar electronic computing device. The computer system or similar electronic device manipulates and transforms data represented as electronic quantities within

the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission, or display devices.

EXEMPLARY HARDWARE UPON WHICH EMBODIMENTS OF THE PRESENT INVENTION MAY BE IMPLEMENTED

Refer now to Figure 1 which illustrates an exemplary computer system 100 upon which embodiments of the present invention may be practiced. In general, computer system 100 comprises bus 110 for communicating information, processor 101 coupled with bus 110 for processing information and instructions, random access (volatile) memory (RAM) 102 coupled with bus 110 for storing information and instructions for processor 101, read-only (non-volatile) memory (ROM) 103 coupled with bus 110 for storing static information and instructions for processor 101, data storage device 104 such as a magnetic or optical disk and disk drive coupled with bus 110 for storing information and instructions.

In one embodiment, computer system 100 comprises an optional user output device such as display device 105 coupled to bus 110 for displaying information to the computer user, an optional user input device such as alphanumeric input device 106 including alphanumeric and function keys coupled to bus 110 for communicating information and command selections to

processor 101, and an optional user input device such as cursor control device 107 coupled to bus 110 for communicating user input information and command selections to processor 101. Furthermore, an optional input/output (I/O) device 108 is used to couple computer system 100 onto, for example, a network.

Display device 105 utilized with computer system 100 may be a liquid crystal device, cathode ray tube, or other display device suitable for creating graphic images and alphanumeric characters recognizable to the user. Cursor control device 107 allows the computer user to dynamically signal the two-dimensional movement of a visible symbol (pointer) on a display screen of display device 105. Many implementations of the cursor control device are known in the art including a trackball, mouse, joystick or special keys on alphanumeric input device 106 capable of signaling movement of a given direction or manner of displacement. It is to be appreciated that the cursor control 107 also may be directed and/or activated via input from the keyboard using special keys and key sequence commands. Alternatively, the cursor may be directed and/or activated via input from a number of specially adapted cursor directing devices.

Referring now to Figure 2, a block diagram of a system 200 upon which the present invention may be practiced is shown. As depicted in Figure 2, system 200 includes managed network 210, network management system

220, network discovery database 230, network simulator 240, device
personality database 250, and simulated network 260. In one embodiment,
network simulator 240, in conjunction with network discovery database 230 and
device personality database 250 is operable to perform a method for
5 generating simulated network 260.

In one embodiment, network management system 220, network
discovery database 230, network simulator 240, device personality database
250, and simulated network 260 are comprised within a single computer
10 system (e.g., computer system 100 of Figure 1). In another embodiment,
network management system 220 is comprised within one computer system
and network discovery database 230, network simulator 240, device personality
database 250, and simulated network 260 are comprised within a second
computer system. In another embodiment, network management system 220,
15 network discovery database 230, network simulator 240, device personality
database 250, and simulated network 260 are dispersed across a distributed
computer network. In one embodiment, the components of system 200
communicate via the communications protocols of system 200. For example,
network management system 250 may communicate with managed network
20 210 and network simulator 240 via simple network management protocol
(SNMP).

Referring still to Figure 2, in accordance with various embodiments, system 200 includes well-known network technologies. For example, system 200 can be implemented using LAN technologies (e.g., Ethernet), the Internet, or other wired or wireless network technologies. The communications links
5 between managed network 210, network management system 220, network discovery database 230, network simulator 240, device personality database 250, and simulated network 260 can be implemented using, for example, a telephone circuit, communications cable, optical cable, wireless link, or the like.

10

In one embodiment, managed network 210 is an actual distributed computer network including a plurality of devices. In one embodiment, the devices include routers and switches. However, it should be appreciated that managed network 210 may include other devices for use in computer
15 networking, such as hubs, or other computing devices, such as servers, computer systems, or voice over Internet protocol (VOIP) telephones.

Network management system 220 is coupled to managed network 210 for managing the configurations, security and health of managed network 210
20 while monitoring many other aspects including the performance of managed network 210. The network management system 220 is also coupled to network discovery database 230 for managed network 210. A network discovery database includes device information and connectivity information for

devices of a managed network. In one embodiment, network management system 220 automatically compiles network discovery database 230 in accordance with monitoring managed network 210. In one embodiment, network discovery database 230 is referred to as an ani.db file.

5

In one embodiment, network simulator 240, which is coupled to network management system 220, network discovery database 230, device discovery database 250 and simulated network 260, is a software application for testing network management system 220 by presenting simulated network 260 to
10 network management system 220. In one embodiment, simulated network 260 is a build file that includes information regarding devices and connections of a network such that network management system 220 operates as if it were connected to an actual network. In one embodiment, network simulator 240 is operable to perform a method for generating simulated network 260 based on
15 managed network 210 (e.g., process 300 of Figures 3A and 3B).

Device personality database 250, which is coupled to network simulator 240, stores the characteristics for a particular type of device. In one embodiment, device personality database 250 includes a plurality of
20 personalities for different device types and operating systems. In one embodiment, a personality is defined by a system object identifier (often referred to as sysobjID), which includes, among many other attributes, a device type and an operating system name and version. For example, a particular type

of router will have at least one personality for each operating system that can operate on the router. A personality includes static information describing a device, such as the device type, the device operating system version, and device interface information. The device interface information includes the
5 number of ports for the device and the type of ports.

A METHOD FOR GENERATING A SIMULATED NETWORK BASED ON AN ACTUAL MANAGED NETWORK

10 Figures 3A and 3B are a flow chart illustrating a process 300 for generating a simulated network based on a managed network, in accordance with one embodiment of the present invention. In one embodiment, process 300 is carried out by processors and electrical components under the control of computer readable and computer executable instructions. The computer
15 readable and computer executable instructions reside, for example, in data storage features such as computer usable volatile and non-volatile memory (e.g., volatile memory 102 and non-volatile memory 103 of Figure 1). However, the computer readable and computer executable instructions may reside in any type of computer readable medium. Although specific steps are disclosed in
20 process 300, such steps are exemplary. That is, the embodiments of the present invention are well suited to performing various other steps or variations of the steps recited in Figures 3A and 3B. In one embodiment, process 300 is performed by network simulator 240 of Figure 2.

With reference to Figure 3A, at step 305 of process 300, a network discovery database (e.g., network discovery database 230 of Figure 2) for a managed network is accessed, wherein the network discovery database includes device information and connectivity information for a plurality of devices of the managed network. In one embodiment, the network discovery database is automatically compiled by a network management system (e.g., network management system 220 of Figure 2). In one embodiment, the plurality of devices includes routers and switches. However, it should be appreciated that other devices for use in computer networking, such as hubs, or other computing devices, such as servers, computer systems, or VOIP telephones.

At step 310, device attributes for a device are extracted from the network discovery database. In one embodiment, the device attributes include a Management Information Base (MIB) for the device. In one embodiment, the device attributes include a device type, network connectivity information, a device configuration, and a device operating system name and version. The device type identifies the type of device, such as a router, switch or other computing device. In one embodiment, the device type is identified by manufacturer model number. The network connectivity information describes the connections and interfaces of the device. In one embodiment, the network connectivity information is a neighbor discovery protocol (NDP) table. A NDP table includes connection information for the particular device, specifically the

identification of all other devices connected to the device. In one embodiment, the NDP table is a Cisco discovery protocol (CDP) table. However, it should be appreciated that NDP table can be any information identifying neighboring devices. The device configuration identifies specific attributes for the particular device, such as line speeds and other configuration information. The device operating system identifies the operating system type and version, such as Cisco Internet Operating System (IOS) version 12.3.

In one embodiment, the device attributes also include event information, such as traps, system logs, and fault log for the device. A trap is a preconfigured rule that activates an alarm upon the occurrence of an event. A system log is a log on the device that logs information upon the occurrence of an event. A fault log includes reported faults of the device.

At step 315, it is determined whether a personality exists for the device. In one embodiment, the system object identifier of the device, which includes the device type and device operating system, is compared to personalities of a device personality database (e.g., device personality database 250 of Figure 2). A personality describes the characteristics for a particular type of device running a particular operating system. For example, the personality includes device interface information, such as the number of ports for the device and the type of ports.

If it is determined that a personality for the device exists, process 300 proceeds to step 320, and the device attributes are attached to the personality. Attaching the device attributes to the personality provides a complete description of the device such that a network simulator (e.g., network simulator 240 of Figure 2) can simulate the device.

Alternatively, if it is determined that a personality for the device does not exist, process 300 proceeds to step 325. At step 325, a personality is created for the device. In one embodiment, the personality is created by accessing the device personality database and selecting a personality that is closely related to the system object identifier of the device. In one embodiment, personalities with the closest model number are first selected. Next, the personality with the closest operating system is selected from the selected personalities. For example, if the device is a 3620 router running Cisco IOS version 11.2, and the closest model number is for a 3600 router, the personalities for the 3600 router are selected. The personality for the 3600 router running Cisco IOS version 11.2 is selected as the personality. Once a personality for the device is created, process 300 proceeds to step 320, where the device attributes are attached to the personality.

20

With reference to Figure 3B, process 300 proceeds to step 330. At step 330, the device attributes and personality for the device are written to a build file. The build file includes all information necessary for a network simulator to

simulate a network for a network management system. In other words, the build file includes information regarding devices and connections of a network such that a network management system operates as if it were connected to an actual network. In one embodiment, the build file is an ASCII file.

5

Figures 4A and 4B are tables 400 and 450 of an exemplary build file, in accordance with one embodiment of the present invention. Table 400 of Figure 4A includes information describing the configuration of exemplary devices of a managed network. For example, device 1 is a 2500 router running on IOS
10 operating system version 10.5. It should be appreciated that any information used for creating a simulated network may be included in table 400, and table 400 is not limited to the present embodiment. With reference to Figure 4B, table 450 is an interface table for device 3 of Figure 4A, including interface identification, connection type, current status (e.g., up or down) and the
15 neighbor device. For example, interface 0 of device 3 is a 10 MB Ethernet connection, currently up, and is connected to device 2. It should be appreciated that each device described in table 400 has an associated interface table describing the neighbor connections for the device.

20

With reference to Figure 3B, at step 335, the event information for the device is configured. In one embodiment, the traps, system logs, and fault logs for the device are configured. The build file is updated to include the event information. By including the event information, an accurate representation of

an actual device of a managed network can be simulated, including all operational history of the device.

At step 340, it is determined whether there are any more devices within the network discovery database that have not been accessed. If there are more devices to access, process 300 accesses the next device, as shown at step 345. Process 500 then proceeds to step 310. Alternatively, if all devices of the network topology have been accessed, process 300 ends.

The described embodiments of the present invention provide a method for generating a simulated network based on a managed network. By accessing a network discovery database associated with the managed network, a simulated network is automatically created, thereby obviating the need for manually coding a simulated network. The simulated network may be used for testing the functionality, reliability, scalability, co-existence and performance of the network management system.

Various embodiments of the present invention, a method for generating a simulated network, are thus described. While the present invention has been described in particular embodiments, it should be appreciated that the present invention should not be construed as limited by such embodiments, but rather construed according to the below claims.